5T982x / 5T989x FAQ

APPLICATION NOTE AN-238

Q1. What controls the Programmable Clock 5T989x//2x VCO operating frequency?
A1. The VCO frequency is controlled by the REF clock frequency, the FB clock frequency, the FB Divide-By settings (Bits 48-51), and the proper selection of the VCO frequency range (Bit 60).

Q2. What is the size of the Programmable Clock programmable skew step?
A2. VCO period divided by 16.

Q3. Can Programmable Clock 5T989x//2x be put in a JTAG chain with FPGAs?
A3. Yes, the 5T989x//2x supports standard JTAG.

Q4. How long does it take to restore and save instructions?
A4. $T_{RESTORE} = 1.23x10^9 / F_{VCO}$ (ms)
$T_{SAVE} = 3.09x10^9 / F_{VCO} + 52$ (ms)

Q5. Can I use the skew controls in PLL bypass mode?
A5. The skew controls remain functional in PLL bypass mode but the skew timing values are fixed and not a function of the input frequency.

Q6. What should I do with unused outputs?
A6. The unused output can be unconnected. Individual unused outputs should be disabled using either the external $\overline{nSOE}$ pins or the programming bits (Bits 52-56). The disabled outputs can either be tri-stated or gated high/low depending on the OMODE pin, and the positive/negative edge control (Bit 58) according to the datasheet.

Q7. What happens to the Programmable Clock I/Os during programming?
A7. Once the part has been fully programmed, the I/Os will switch to the new settings asynchronously. The part will not react to the new settings until the programming process is completely finished.

Q8. What happens if the reference clock is removed?
A8. Removing the REF clock will cause the part to lose lock, and the VCO will ramp down to its minimum frequency (which will be in the range of 20 to 50MHz). The part can be programmed in this condition.

Q9. What models are available for the Programmable Clock?
A9. IBIS and Hspice models of the I/O are available.

Q10. Is it possible to re-program Programmable Clocks on the fly without affecting the EEPROM content?
A10. Yes, the ProgWrite programming instruction re-programs the Programmable Clock without affecting the EEPROM contents. The EEPROM contents will only be affected by the ProgSave programming instruction.

Q11. What would be the impact on the outputs (glitch, phase offset, stabilization, etc.…) if the Programmable Clock were to be re-programmed on the fly?

A11. It depends on which configuration settings are changing with the new programming. The output disable/enable bits (Bits 52-56) function the same as the external pins. The outputs will be synchronously enabled/disabled by the bits. All other configured settings that affect the outputs will asynchronously change the outputs. The outputs will move freely according to the new settings, so they may glitch, and the PLL will lose lock and have to re-stabilize for the new setting. It may take up to 1ms for the device to relock.

Q12. Is there a SVF file provided for JTAG programming?

A12. Yes. The IDT Software Kit can generate SVF files.

Q13. Are the JTAG/ I^2C pins 3.3V tolerant?

A13. The TDI/SDA and TCK/SCLK pins are 3.3V tolerant. TDO/ADDR1 and TMS/ADDR0 are not 3.3 V tolerant because of the tri-level nature of the ADDR0/ADDR1. It is important to note that I^2C users could use a 3.3V bus (SDA and SCLK).

Q14. What are the sequences of events during power-up?

A14. On power-up of the 5T989x//2x parts, an automatic restore is performed to load the EEPROM contents into the chip's volatile programming registers. Before the auto-restore completes, the chip's volatile programming registers contain the value '0' for all bits 95:0, and this will be the configuration of the chip as the power is coming up. During this time, the PLL is held at min frequency and will not achieve phase lock until after the auto-restore completes. Once the auto-restore completes, the chip will reconfigure to the new settings and attempt to achieve phase lock to the REF CLK.

Q15. What are the sequences of events after re-programming?

A15. In order to re-program the device, the chip must be powered up and the auto-restore completed. Then a ProgWrite instruction must be issued from JTAG or I^2C. The device's volatile programming registers will be overwritten with the new data only after the entire instruction has completed. The new configuration settings will then take effect asynchronously.

Q16. Should I reset the device after programming?

A16. The device should not need to be reset if the datasheet instructions are followed regarding the $\overline{\text{PLLEN}}$ and the programming interface. However, it will need to be reset if an EEPROM instruction (ProgSave or ProgRestore) instruction is issued when the PLL is turned off ($\overline{\text{PLLEN}} = V_{DD}$).

Q17. Under what conditions can the auto-restore fail? What happens when this process fails?

A17. The only condition in which the auto-restore will fail once initiated is if there is an EEPROM failure. If an EEPROM goes bad or if the Programmable Clock chip is being operated outside of its V_{DD} specifications, then the auto-restore may fail. In this case, no programming instructions will function, including ProgWrite and ProgRead.

To be initiated, the auto-restore requires that the $\overline{\text{PD}}$ input be held high (the part must not be in power-down mode on power-up). If the part is powered-up with $\overline{\text{PD}}$ low, then the auto-restore will not begin and no programming instruction will be accepted by the device. If the auto-restore fails and the user needs to use the programming interface, then the device must be power-cycle to re-try the auto-restore. If auto-restore continues to fail, then something is wrong with the device and it needs to be replaced.

Q18. What happens if a user writes to the I^2C or JTAG pins while the chip is doing auto-restore or performing previous instructions such as PROGSAVE?

A18. If a ProgRestore or ProgSave is already in progress, any new JTAG or I^2C programming instruction will be ignored. The I^2C interface will not respond to its address while the EEPROM is being accessed, and the JTAG controller will not recognize a new programming instruction until the EEPROM controller indicates the previous process is complete.

Q19. Is there a way to verify that the SVF file will run correctly with the Programmable Clock?

A19. One way to do this is to make sure there are ample clock cycles for RUNTEST between instructions, especially the ProgSave and ProgRestore instructions to the EEPROM. This should be based on the following equation:

$T_{RESTORE} = 1.23x10^9 / F_{VCO}$ (ms)

$T_{SAVE} = 3.09x10^9 / F_{VCO} + 52$ (ms)

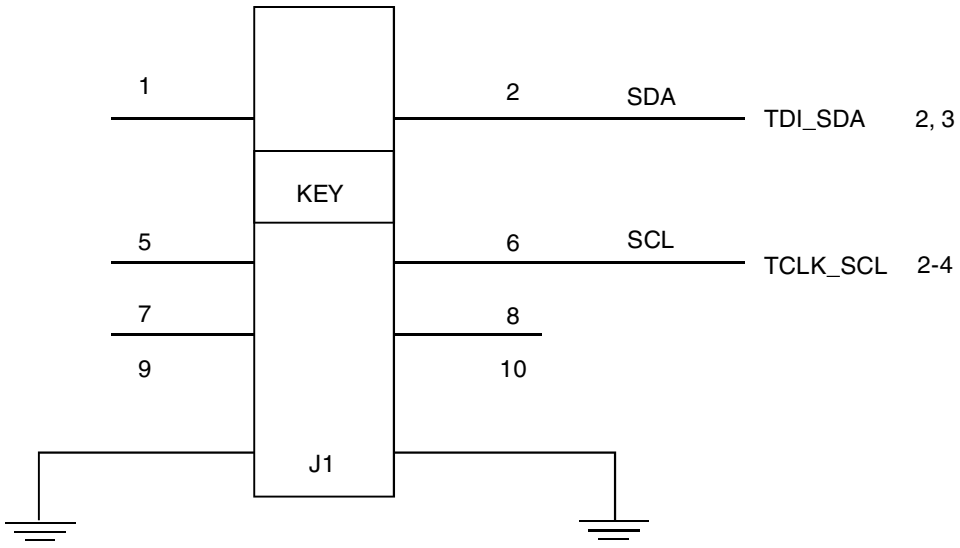
Q20. How can I use JTAG BYPASS instruction to read from EEPROM using SVF file?

A20. BYPASS mode places a device in non-boundary scan testing mode while the pins remain unaffected, and the device should be operating normally. It places the DR shift register in a pass through mode, with an effective register bit length of one. This mode is not normally used for device configuration. One of the PRIVATE (i.e. ProgWrite, ProgRead, ProgRestore, and ProgSav) instructions should be shifted into the IR register (the SVF SIR command) to place the device in a mode that will allow shifting in of the frequency configuration data using shift DR mode (the SVF SDR command). After that, a BYPASS instruction would be shifted into the IR register (the SVF SIR command) to place the part back into normal mode. If you leave the device in configuration mode, it might shift in bad data if there is any noise on the signals.

Q21. What type of I^2C connector and cable is used for the evaluation board and programmer kit?

A21. Part Number:
IDT STG Library Part Number: 52-492-000
MEC1-105-02-S-D-LC (CONN,SM,2 x 5,Edge-Card Skt,1.0mm PT,STR)

Drawing:
http://www.samtec.com/ftppub/cpdf/MEC1-1XX-XX-X-D-XX-XX-MKT.pdf



IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.